UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉNICA SUPERIOR

Doble Grado en Ingeniería Informática y Matemáticas

TRABAJO FIN DE GRADO

Bandidos multi-brazo en sistemas de recomendación

Autor: Esther López Ramos
Tutor: Javier Sanz-Cruzado Puig
Ponente: Pablo Castells Azpilicueta

junio 2019

Esther López Ramos
Bandidos multi-brazo en sistemas de recomendación

Esther López Ramos

IMPRESO EN ESPAÑA – PRINTED IN SPAIN

Una mente necesita de libros igual que una espada
de una piedra de amolar, para conservar el filo.

George R.R. Martin

Agradecimientos

Con este trabajo cierro una de las etapas más importantes de mi vida. Por ello me gustaría agradecer a todas aquellas personas que me han ayudado y apoyado durante estos cinco años. En primer lugar quisiera dar las gracias a todos mis compañeros del doble grado por la ayuda compartiendo apuntes y dudas. En especial a Antonio, Carlos y Javi no solo por su ayuda con las prácticas y con los exámenes de matemáticas, sino también por compartir los ratos libres. En segundo lugar quiero dar las gracias a mi familia por su apoyo y paciencia, ya que sin ellos jamás habría logrado mis objetivos. Por último quisiera agradecer a Pablo y a Javier y en general a todo el grupo de Recuperación de Información por haberme brindado la oportunidad de hacer este trabajo y haberme acogido en su laboratorio.

Resumen

En los últimos años los sistemas de recomendación se han convertido en una herramienta fundamental ampliamente utilizada por gran cantidad de servicios muy diversos, tales como comercios electrónicos, redes sociales o plataformas de *streaming*. Este crecimiento se debe, en parte, a que los sistemas de recomendación proporcionan grandes ventajas tanto para el usuario, en tanto que mejoran su experiencia proporcionándole ideas sobre qué contenido consumir, como para los proveedores del servicio, ya que favorecen el consumo por parte de los usuarios y les ayudan a descubrir productos que de otra forma no habrían conocido.

Este creciente auge de los sistemas de recomendación ha promovido el desarrollo de técnicas y herramientas *software* que persiguen la satisfacción por parte del usuario centrándose en mejorar las recomendaciones. Estas técnicas tienen un gran fundamento en el campo del Aprendizaje supervisado, puesto que la tarea de recomendación se ha entendido de manera tradicional como un problema de predicción. El hecho de tratar las recomendaciones como una predicción de la puntuación que el usuario daría a cada uno de los ítems que desconoce, supone entender la recomendación desde un punto de vista estático de un único paso en el que se busca el acierto inmediato sin tener en cuenta las consecuencias a largo plazo.

Este trabajo aborda una nueva perspectiva en cuanto la recomendación que supone romper con este enfoque más tradicional: el Aprendizaje por Refuerzo. Mediante este nuevo enfoque, la recomendación pasa a entenderse como un proceso de aprendizaje continuo en el que las interacciones del usuario con las recomendaciones se aprovechan para adquirir nuevo conocimiento y mejorar a futuro. Por ello, se admite un sacrificio del acierto inmediato en pro de obtener un mayor beneficio a largo plazo. Esta idea de proceso interactivo entre usuario y sistema resulta mucho más natural que el enfoque estático más tradicional inspirado por el Aprendizaje supervisado. Más concretamente en este trabajo se profundiza en técnicas conocidas como bandidos multi-brazo, que tienen base en la teoría de la probabilidad, aplicadas a la tarea de la recomendación no personalizada. Dentro de los bandidos multi-brazo, se estudian los algoritmos de ε-Greedy, Upper Confidence Bound y Thompson Sampling, comparándolos entre sí y con algoritmos clásicos de recomendación no personalizada en diferentes conjuntos de datos. Los resultados obtenidos indican que en todos los conjuntos de datos existen configuraciones de al menos uno de los algoritmos anteriores que mejoran el acierto con respecto a los algoritmos de recomendación sin refuerzo.

Palabras clave

Aprendizaje por refuerzo, sistemas de recomendación, bandidos multi-brazo

Abstract

Over the last years, recommender systems have become a widely used tool by many diverse online services such as e-commerce platforms, social networks, and streaming services. For the most part, this growth has been due to the great advantages recommender systems offer to both users, as long as they improve their experience by offering new ideas about what content they might like, and service providers because they help the users find new content which leads to an increase on product consumption.

This growth on recommender systems has promoted the development of new software tools and techniques which pursuit user satisfaction focusing on the improvement of recommendations. These techniques are based in the field of Supervised learning because traditionally, the task of recommendation has been treated as a prediction problem. Treating recommendations as a prediction of the rating the user would give to each unknown item implies understanding the recommendation as a static approach where only the immediate success is considered without taking into account long-term consequences.

In this dissertation, we address a new perspective that breaks this traditional way of understanding recommendation: Reinforcement Learning. With this new approach, the recommendation is understood as a continuous learning process in which the user feedback is considered to get new knowledge and improve future recommendations. Now we sacrifice the immediate reward with the aim of getting more benefit in the long-term. This idea of an interactive process between the user and the system is much more natural than the traditional static approach inspired by Supervised learning. More specifically, we focus on techniques known as multi-armed bandits, which are based on Probability Theory, applied to the task of non-personalised recommendation. In the multi-armed bandit context, we study the algorithms ε-Greedy, Upper Confidence Bound and Thompson Sampling, comparing them to each other and to classic non-personalised recommender algorithms in diverse datasets. The results show that in every dataset, there are configurations of at least one of the mentioned algorithms that provide an improvement to the success with respect to algorithms without reinforcement.

Keywords

Reinforcement learning, recommender systems, multi-armed bandits

ÍNDICE

Listas

Lista de ecuaciones

Lista de figuras

Lista de tablas

1

Introducción

Los sistemas de recomendación han supuesto un cambio en cuanto a cómo entendemos a día de hoy la gran mayoría de servicios de la Web. Pese a la gran cantidad de herramientas *software* existentes, el problema de la recomendación sigue teniendo aún un largo recorrido, puesto que es una tarea difícil, si no imposible, de resolver de manera exacta. Tradicionalmente, la tarea de recomendación se ha entendido como un proceso de un único paso, es decir, el sistema recomienda al usuario una serie de objetos o ítems y este los evalúa y devuelve su *feedback* al sistema. No obstante, resulta más natural entenderla como un proceso interactivo y continuo, en el que el sistema aprovecha el *feedback* del usuario para aprender sobre él y mejorar las futuras recomendaciones. Este trabajo se centra en explorar esta nueva forma de entender la recomendación desde la perspectiva del aprendizaje por refuerzo, más concretamente, utilizando métodos conocidos como bandidos multi-brazo.

1.1. Motivación

La principal razón que motiva el presente trabajo es que la perspectiva del aprendizaje por refuerzo resulta mucho más conveniente y natural para la recomendación que los enfoques estáticos más tradicionales. Además, la investigación en cuanto a la aplicación del aprendizaje activo a los sistemas de recomendación es relativamente reciente, con mucho margen por explorar y con creciente interés. Si bien existen líneas de investigación abiertas en la aplicación de esta perspectiva en cualquiera de sus variantes a la tarea de la recomendación, hay todavía muchas dimensiones por explorar, como puede ser la recomendación de contactos en redes sociales o la consideración de otras métricas complementarias al acierto.

El problema de los bandidos multi-brazo supone un problema clásico de la teoría de la probabilidad que se presta a una clara formalización y sobre el que ya existe un amplio desarrollo teórico [Sutton and Barto, 2018]. Asimismo, la búsqueda de un balance entre exploración y exploración es un problema recurrente en muchas áreas de la ingeniería informática. En concreto, tiene una aplicación clara y directa en mútiples facetas de la recuperación de la información, como pueden ser los test A/B [Scott, 2015, Hill et al., 2017], por lo que es un área fértil en conexiones con otros problemas del campo.

1.2. Objetivos

Teniendo en cuenta los puntos expuestos anteriormente que motivan este trabajo, se plantea el objetivo principal de estudiar la perspectiva del aprendizaje activo sobre los sistemas de recomendación. Dicho objetivo puede subdividirse en los siguientes.

- Estudiar métodos generales de aprendizaje por refuerzo profundizando en la teoría de bandidos multi-brazo y en sus estrategias para obtener un balance entre exploración y explotación.
- Aplicar los métodos anteriores a la tarea de recomendación tomando una perspectiva iterativa en la que el *feedback* que proporciona el usuario se añade a la base de conocimiento para maximizar el acierto a largo plazo.
- Implementar algoritmos de bandidos multi-brazo, concretamente ε-Greedy, Upper Confidence Bound y Thompson Sampling.
- Entender cómo varían los bandidos multi-brazo con la inicialización de sus parámetros, y qué configuraciones funcionan mejor en su aplicación a los sistemas de recomendación en diferentes dominios: redes sociales, música y cine.
- Considerar diferentes funciones objetivo además del acierto, como puede ser el descubrimiento de nuevos ítems por parte del usuario.

1.3. Metodología

Para llevar a cabo los objetivos mencionados en la sección anterior, el desarrollo de este trabajo se ha estructurado en tres fases claramente diferenciadas. La primera fase corresponde con una lectura y estudio de la literatura. En primer lugar se han consultado libros y artículos que proporcionan una introducción a la tarea de la recomendación y a los algoritmos más extendidos para resolverla. Posteriormente se ha estudiado la perspectiva del aprendizaje por refuerzo en general y aplicada a los sistemas de recomendación, centrándose en los algoritmos de bandidos multi-brazo. Tras esta fase de estudio se ha procedido a realizar una implementación en Python. En primer lugar se ha implementado una plataforma sencilla de ejecución dedicada a realizar el ciclo iterativo de recomendación mediante el cual el sistema va recomendando ítems a los usuarios y actualizando su información. Posteriormente se ha procedido a implementar los tres algoritmos de bandidos multi-brazo escogidos, adaptándolos a la recomendación. En última instancia, se han realizado diversos experimentos con tres conjuntos de datos. Primero se han estudiado los conjuntos de datos en profundidad y se han transformado a un formato común para ejecutar los algoritmos. A continuación se ha procedido con los experimentos.

1.4. Estructura del documento

El documento se estructura en los siguientes cinco capítulos:

1. **Introducción**. En ella se exponen las principales razones que motivan la elaboración de este trabajo, los objetivos que de ellas derivan y el plan general de desarrollo que se ha seguido para llevarlos a cabo.
2. **Estado del arte**. En este capítulo se describen de forma general los problemas que se tratan en este trabajo. Se comienza con una introducción a la tarea de recomendación. Posteriormente se describe el papel del aprendizaje por refuerzo sobre esta tarea y se exponen las dos principales líneas de investigación abiertas, centrándose en las técnicas de bandidos multi-brazo.
3. **Planteamiento y algoritmia**. Este capítulo tiene dos objetivos principales. El primero es adaptar la teoría de bandidos multi-brazo a los sistemas de recomendación. El segundo es describir de manera profunda los tres algoritmos escogidos para desarrollar en este trabajo.
4. **Experimentos**. Sobre tres conjuntos de datos de diversas características, se realizan experimentos con los tres algoritmos con el objetivo de determinar las mejores configuraciones para cada uno de ellos y compararlos entre sí.
5. **Conclusiones**. En este capítulo se resumen los resultados de los experimentos y se exponen las futuras líneas de trabajo.

2

Estado del arte

El aumento en los últimos años de la popularidad y la utilidad de los Sistemas de Recomendación ha desembocado en el desarrollo de nuevas técnicas y variaciones de la tarea original de recomendación. En este capítulo se persigue, en primer lugar, ofrecer al lector poco familiarizado con esta tarea una visión general de los elementos que la componen y sus objetivos. Posteriormente, se hace una introducción sobre el problema de aprendizaje por refuerzo, exponiendo las dos líneas principales de investigación abiertas actualmente. Por último se ofrece un resumen de las técnicas de bandidos multi-brazo más utilizadas en la literatura y su aplicación al contexto de los Sistemas de Recomendación.

2.1. Sistemas de Recomendación

El auge en los últimos años del comercio electrónico, las redes sociales y los servicios en *streaming* ha propiciado el desarrollo de técnicas *software* cuyo objetivo es enriquecer la experiencia de los usuarios proporcionándoles sugerencias sobre los productos ofertados. Estas técnicas se conocen como Sistemas de Recomendación [Ricci et al., 2011].

Un sistema de recomendación cuenta con dos conjuntos básicos: el conjunto de usuarios, a partir de ahora llamado U y el conjunto de ítems, denotado por I. El objetivo de un sistema de recomendación es proporcionar a cada usuario un ranking de ítems, es decir, una lista ordenada de mayor a menor según la estimación que el sistema hace sobre la utilidad de los ítems para el usuario. Preferiblemente, estos ítems serán desconocidos por el usuario. Se distinguen dos tipos de recomendaciones:

1. **Recomendación personalizada**. Considera las preferencias del usuario a la hora de calcular las recomendaciones, por tanto los rankings son diferentes para cada usuario. Generalmente los gustos de los usuarios se computan como valores escalares que se denominan *ratings* o puntuaciones, donde los valores más altos indican preferencia. La recomendación consiste en predecir la puntuación de los ítems no valorados por el usuario y devolver los n ítems con más puntuación estimada. Algunos de los algoritmos más famosos para este tipo de recomendación son *K-Nearest Neighbours (KNN)* o K-Vecinos más próximos [Ricci et al., 2011, chap. 2] y *Matrix factorization* o Factorización de matrices [Ricci

et al., 2011, chap. 5].

2. **Recomendación no personalizada**. Efectúa la misma recomendación para todos los usuarios basándose en aspectos generales como la popularidad de los ítems. Es especialmente útil cuando un usuario es nuevo en el sistema y no se dispone de información previa sobre sus preferencias. Para este tipo de recomendaciones se tienen en cuenta de nuevo las puntuaciones de los usuarios a los ítems para identificar cuáles son los ítems mejor valorados, por ejemplo calculando la media aritmética entre todas las puntuaciones efectuadas sobre un ítem.

En la figura 2.1 se ilustra de forma esquemática el comportamiento de un sistema de recomendación.

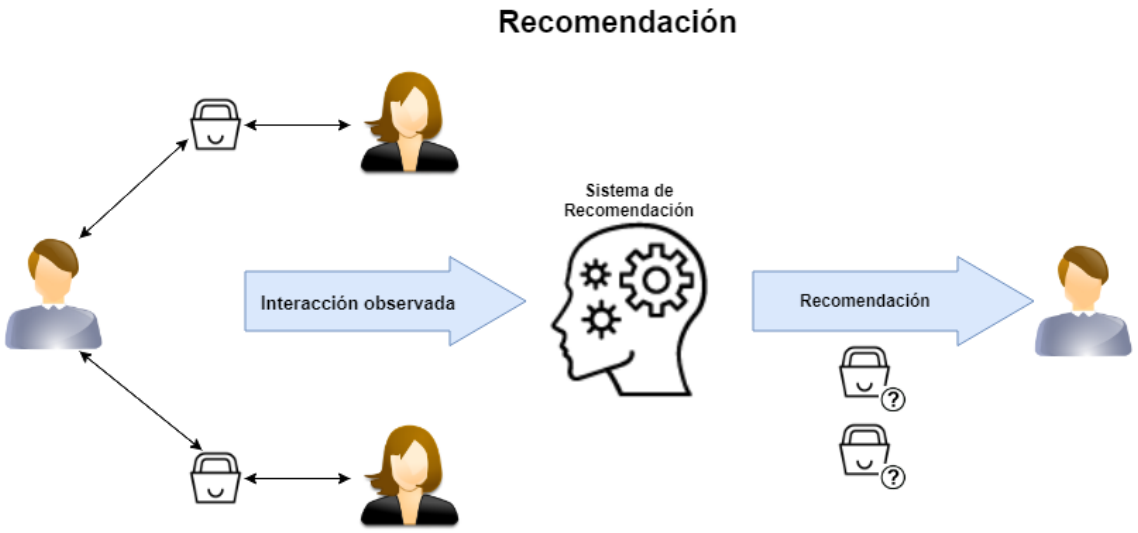

Figura 2.1: Esquema del funcionamiento de un sistema de recomendación

Para comprobar la efectividad de un sistema de recomendación se plantean principalmente dos tipos de evaluación:

1. **Evaluación *offline***. Este enfoque está muy inspirado en las técnicas de evaluación propias del Aprendizaje supervisado. Consiste en tomar un conjunto de valoraciones de usuarios a ítems conocido, referido habitualmente en la literatura como conjunto de *ratings* y denotado por R y dividirlo en dos subconjuntos de entrenamiento, R_{train}, y test, R_{test}. De esta forma para las parejas $(u, i) \in R \subset U \times I$ se define la función $r(u, i)$ que toma valores en $\mathbb{R}$. Las puntuaciones del conjunto R_{test} se ocultan al sistema y su tarea es estimarlas empleando la información disponible en R_{train}. Por tanto, la tarea de recomendación se reduce a predecir los valores de $r(u, i)$ para $(u, i) \in R_{test}$. La estimación que el sistema hace sobre estos valores será denotada por $\hat{r}(u, i)$. Una vez obtenidas estas predicciones, se comparan con los valores reales de R_{test} atendiendo a diferentes métricas que también tienen inspiración en el Aprendizaje supervisado, siendo las más utilizadas precisión y

recall [Ricci et al., 2011].

2. **Evaluación *online*** Este enfoque implica la evaluación del sistema de recomendación en un entorno en producción, por lo que supone una tarea complicada. En primer lugar, se necesita un sistema de confianza con un elevado número de usuarios. En segundo lugar, con el objetivo de no empobrecer la experiencia del usuario, debería combinarse el algoritmo de recomendación existente en el sistema con la nueva propuesta, lo que implicaría dividir la carga de peticiones al sistema de recomendación. Por último, cabe la posibilidad de que muchos de los usuarios no respondan a las recomendaciones propuestas, lo que dificulta enormemente la evaluación de los algoritmos. Estas técnicas son comúnmente conocidas como Test A/B.

Puesto que el enfoque más tradicional de los Sistemas de Recomendación consiste en predecir las puntuaciones que un usuario daría a los ítems no valorados, muchos de los algoritmos más utilizados actualmente están influenciados por algoritmos clásicos de Aprendizaje supervisado y por tanto tienen un enfoque estático de la recomendación, es decir, se asume que las puntuaciones dadas por un usuario no cambian con el tiempo. Asimismo se toman muchas técnicas del campo de la Recuperación de la información o *Information Retrieval*, ya que la tarea de recomendación se puede entender como una de búsqueda en la que no hay consulta. Por estos motivos la recomendación se entiende como un proceso de un solo paso en el que se muestra el ranking al usuario y este responde a la recomendación obteniendo un grado mayor o menor de satisfacción de sus gustos.

En este trabajo se abordarán estrategias que van más allá de este enfoque estático de los algoritmos de recomendación más tradicionales, entendiendo que la recomendación es un proceso de aprendizaje iterativo, en el que el sistema y los usuarios interactúan repetidas veces a lo largo del tiempo. En este proceso, la retroalimentación que aporta el usuario tras recibir la recomendación es empleada por el sistema para adquirir nuevo conocimiento. Esta perspectiva plantea un nuevo problema: la recomendación que maximiza la probabilidad de acierto inmediata no es necesariamente la que proporciona más información sobre el usuario al sistema. El reto es pues combinar dos objetivos, uno a corto plazo (explotación) y otro a medio y largo plazo (exploración), buscando maximizar la satisfacción del usuario en una perspectiva temporal amplia. Este es un problema genuino del ámbito llamado aprendizaje por refuerzo.

2.2. Aprendizaje por refuerzo

El aprendizaje por refuerzo (*reinforcement learning*) [Sutton and Barto, 2018] se puede entender como un paradigma del aprendizaje automático. Lleva implícita la idea natural de que un agente en contacto con su entorno, realiza una serie de acciones y estas le producen una respuesta positiva o negativa. Gracias a esa respuesta, el agente es capaz de adquirir conocimiento sobre el medio.

En los problemas clásicos de aprendizaje automático, el objetivo principal es maximizar la recompensa acumulada. El aprendizaje por refuerzo mantiene ese objetivo variando la forma de alcanzarlo. En el aprendizaje por refuerzo existe un agente cuyo objetivo es encontrar un balance entre la exploración y la explotación, es decir, entre interaccionar con el medio realizando una acción desconocida para adquirir nuevo conocimiento y optimizar las decisiones basadas en el conocimiento adquirido para lograr una mayor recompensa. En este contexto, el agente interacciona con el medio en pasos discretos en los que obtiene una observación. Tras esto, el agente escoge una acción entre las posibles, lo que le produce un nuevo estado. Normalmente, el agente es capaz de observar el medio de manera global, aunque puede darse la situación de que solo disponga de información parcial. Este proceso de interacción constante con el medio suele ser prolongado en el tiempo. Por este motivo se asume que la recompensa a corto plazo no sea óptima, sino que se sacrifica el acierto a corto plazo para lograr una mayor recompensa a largo plazo.

Esta idea de proceso interactivo entre el agente y el medio es aplicable de manera natural a los sistemas de recomendación, puesto que gran parte de la información que emplean los diferentes algoritmos para realizar la recomendación procede directamente del usuario, lo que conlleva una interacción constante usuario-sistema. Esta nueva perspectiva rompe con el enfoque más tradicional, donde el problema de recomendación se reduce a un problema de predicción estático. Ahora se considera el enfoque secuencial de la recomendación, lo que permite tener en cuenta el efecto a largo plazo de las recomendaciones. Asimismo, la aplicación del aprendizaje por refuerzo a los sistemas de recomendación resulta útil cuando el sistema no tiene datos suficientes sobre las preferencias del usuario. Esta situación es común cuando, por ejemplo, un usuario llega por primera vez al sistema. Una de las soluciones habituales a este problema es preguntar al usuario sobre sus preferencias, como hacen, por ejemplo, aplicaciones como Linkedin o Twitter. No obstante, no es viable mostrar al usuario todos los ítems de los que dispone el sistema, por lo que la exploración resulta ideal en este contexto. Así pues, la figura 2.1 se ve modificada según la figura 2.2 al tratar la recomendación desde la perspectiva del aprendizaje por refuerzo.

Existen dos líneas principales de investigación actualmente en cuanto a la aplicación del aprendizaje por refuerzo a los sistemas de recomendación se refiere. La primera de ellas consiste en plantearlo como un proceso de decisión de Markov o *Markov Decision Process (MDP)* [Shani et al., 2005]. La segunda línea es en la que se centra este trabajo, que consiste en ver el problema de recomendación como un problema de bandidos multi-brazo o *Multi-armed bandits (MAB)* [Sutton and Barto, 2018].

2.3. Procesos de decisión de Markov

Un proceso de decisión de Markov [Littman et al., 1995], es una cuádrupla (S, A, P, R), donde S es un conjunto finito de estados, A es un conjunto finito de acciones, P es una función de probabilidad que, dados dos estados $s_1, s_2 \in S$ y una acción $a \in A$, devuelve la probabilidad de pasar del estado s_1 a s_2

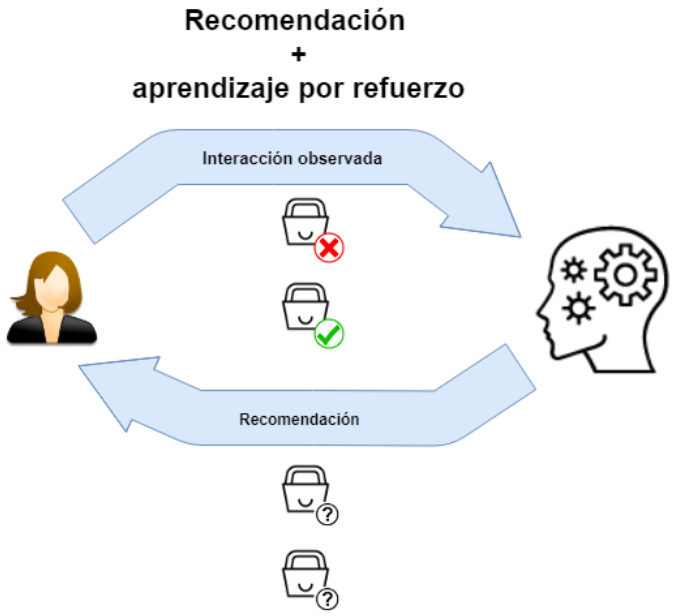

Figura 2.2: Esquema del funcionamiento de un sistema de recomendación con aprendizaje por refuerzo

al realizar la acción a, y R es la recompensa asociada al pasar de s_1 a s_2 mediante a.

En el contexto de este trabajo, el aprendizaje por refuerzo, las probabilidades P o las recompensas R son conocidas y a esta cuádrupla se le añade un agente que es quién toma las decisiones sobre que acción realizar. El objetivo del agente es maximizar la recompensa acumulada. Para ello, observa el entorno y basándose en las observaciones obtenidas, escoge una acción entre las posibles de A y a través de ella interactúa con el entorno. Tras escoger la acción, el agente se mueve a un nuevo estado y recibe una recompensa. Resolver el problema equivale a determinar la secuencia de acciones que produce la mayor recompensa acumulada desde cualquier estado posible.

Aplicado a los sistemas de recomendación, cada estado se corresponde con la información de la que el sistema dispone en un instante de tiempo, cada acción corresponde a recomendar un ítem y, por tanto, la recompensa asociada a cada transición equivale a la aceptación o rechazo de la recomendación por parte del usuario. Para resolver el MDP , en el artículo [Shani et al., 2005] se propone emplear una estrategia de iteración de políticas (*Policy Iteration* [Littman et al., 1995, chap. 4]). En este algoritmo, se proponen una serie de soluciones llamadas políticas que asignan a cada estado una acción y se alterna entre dos fases. En la primera se evalúan las políticas (exploración) y en la segunda se escoge la mejor política y se mejora (explotación). Ambas fases se iteran hasta llegar a una convergencia.

La principal diferencia entre los MDP y los bandidos multi-brazo radica en que en los MDP hay un estado en cada instante de tiempo, mientras que en los bandidos multi-brazo esto no ocurre. No obstante existen estrategias de bandidos multi-brazo que sí tienen en cuenta diferentes característi-

cas que describen tanto al agente como al entorno. Estos bandidos reciben el nombre de bandidos contextuales [Sutton and Barto, 2018, chap. 2].

2.4. Bandidos multi-brazo

El problema de "bandidos multi-brazo" proviene de las máquinas tragaperras. En él un jugador se encuentra frente a un determinado número de máquinas y debe ir accionando los brazos uno a uno con el objetivo de lograr la mayor recompensa. Más formalmente [Sutton and Barto, 2018, chap. 2], un agente debe escoger entre una serie de acciones que se conocen como brazos. Las acciones, al ser seleccionadas reportan un valor numérico llamado recompensa o *reward* que es desconocido por el usuario. Además dicho valor no se mantiene constante, es decir, no se reporta el mismo valor cada vez que se toma la misma decisión, sino que sigue una distribución desconocida. El objetivo del problema de bandidos multi-brazo es tomar una serie de decisiones consecutivas que permitan maximizar la recompensa acumulada, que, en caso de que esas distribuciones se mantengan invariantes a lo largo del tiempo, se traduce en encontrar el brazo o acción cuya distribución tenga la mayor media (de ahora en adelante nos referiremos a esta media como "valor del brazo").

Las estrategias que se siguen para abordar el problema son aproximadas y se basan en efectuar un balance entre la exploración y explotación. En este sentido, el agente puede accionar repetidas veces el brazo que considera hasta el momento que tiene el mayor valor o, alternativamente, puede seleccionar otras acciones para obtener nuevo conocimiento sobre las mismas. Entre los algoritmos más empleados en la literatura, se encuentran ε-Greedy [Sutton and Barto, 2018, chap. 2], *Upper Confidence Bound* (UCB) [Auer et al., 2002] y *Thompson Sampling* [Chapelle and Li, 2011], que serán desarrollados más a fondo en la sección 3.2.

Al aplicar los algoritmos de bandidos multi-brazo a la tarea de recomendación se debe establecer una correspondencia entre los elementos que forman parte de los Sistemas de Recomendación (usuarios, ítems y *ratings*) y los elementos propios de los bandidos multi-brazo (agente, acciones o brazos y sus recompensas). En la mayor parte de la literatura relacionada con el tema, los ítems son considerados como los brazos del bandido y por tanto, accionar un brazo equivale a recomendar un ítem. Este es el enfoque escogido para desarrollar este trabajo y se explicará más en detalle en la sección 3.1. Además se introduce un elemento adicional que habitualmente no está presente en los problemas de bandidos multi-brazo: el usuario. Al tratar cada ítem como un brazo, la tarea del usuario es, por tanto, determinar la recompensa inmediata de la recomendación, es decir, dar una medida de cuán buena le ha resultado la recomendación. Por este motivo, la aplicación de bandidos multi-brazo a los sistemas de recomendación forma parte de lo que se denominan bandidos contextuales [Sutton and Barto, 2018, chap. 2].

En este trabajo, el valor de un brazo será constante para todos los usuarios, lo que proporciona una

estrategia no personalizada. No obstante se han desarrollado en este área modelos y algoritmos más complejos que proporcionan estrategias personalizadas. Algunos de ellos están basados en algoritmos de factorización de matrices [Zhao et al., 2013, Wang et al., 2018, Kawale et al., 2015] y asumen que la recompensa puede descomponerse en factores que dependen de los usuarios y/o los ítems. Otros enfoques proponen agrupar los usuarios y los ítems en *clusters* [Gentile et al., 2014, Li et al., 2010]. Algunos de estos algoritmos aprovechan la disponibilidad de datos adicionales sobre los ítems. Con el fin de ofrecer un primer contacto con el problema de bandidos multi-brazo, en este trabajo se ha optado por implementar algoritmos que proporcionan una estrategia no personalizada. Todos los detalles sobre el planteamiento del problema y los algoritmos empleados se encuentran en el capítulo 3.

2.5. Software de aprendizaje por refuerzo

En lo que respecta a paquetes *software* que proporcionen implementaciones de los algoritmos de bandidos multi-brazo, se han consultado varias. Las más destacables, que cuentan con los algoritmos que se han escogido para este trabajo (ver sección 3.2), son [Ouyang, 2017], que, de las consultadas, es la más completa y la que más algoritmos implementa, y [Galbraith, 2016], que al estar escrita en Python, resulta conveniente pues es el lenguaje escogido para llevar a cabo los experimentos del capítulo 4.

Para poder adaptar estos módulos a la tarea de recomendación es necesario especificar un usuario, que determine la recompensa al proporcionarle un ítem. En estos módulos se definen diferentes distribuciones de probabilidad para la recompensa, encapsuladas en clases. Para poder adaptar la recomendación, habría que generar una nueva clase que devolviese la recompensa asociada a un ítem una vez se ha fijado el usuario. No obstante, los parámetros que reciben estas clases están fijados y no se puede introducir el usuario. La única manera de simular este comportamiento es escoger aleatoriamente un usuario de entre todos los posibles, pero esta selección aleatoria rompe la perspectiva de la recomendación en la que es el usuario a quien se le proporciona un ranking de ítems. Por tanto, estos módulos no son fácilmente adaptables a la recomendación, ya que no es trivial introducir el contexto que proporciona el usuario. Por este motivo se ha optado por una implementación propia.

3

Planteamiento y algoritmia

Este capítulo se centra en formalizar la metodología empleada para adaptar los algoritmos de bandidos multi-brazo al contexto de los sistemas de recomendación. Para ello se emplean dos secciones. La primera sección contextualiza el problema general del aprendizaje por refuerzo a los sistemas de recomendación. En ella, se desarrolla el proceso general seguido en los experimentos del capítulo 4 y las métricas que se utilizan para evaluar la efectividad de los algoritmos. La segunda, corresponde a los algoritmos que se usarán en el resto de capítulos y su filosofía para realizar el balance entre exploración y explotación.

3.1. Planteamiento: bandidos recomendadores

3.1.1. Adaptación de la tarea

Uno de los objetivos del presente trabajo consiste en aplicar algoritmos de bandidos multi-brazo para generar recomendaciones a diferentes usuarios. Los bandidos multi-brazo pueden utilizarse en diferentes dominios, por lo que, en primer lugar, se debe establecer una relación entre el problema que se persigue resolver y la formulación de los bandidos. Como ya se ha detallado en el capítulo 2, un sistema de recomendación cuenta con los siguientes elementos: el conjunto de usuarios, a partir de ahora llamado U y el conjunto de ítems, denotado por I. En este trabajo, para cada pareja (u, i) se define la puntuación (*rating*) como sigue:

$$r\colon R \subset U \times I \to \{0, 1\}$$
$$(u, i) \to r(u, i) = \begin{cases} 1 & \text{en caso contrario} \\ 0 & \text{si a } u \text{ no le gusta } i \end{cases} \tag{3.1}$$

donde el conjunto R representa el conjunto de valoraciones o *ratings*, es decir, los pares (u, i) tales que el valor de $r(u, i)$ es conocido por el sistema. Los valores de $r(u, i)$ desconocidos son estimados para poder realizar la recomendación. Esta estimación de ahora en adelante será denotada por $\hat{r}(u, i)$.

La acción básica de un Sistema de Recomendación es seleccionar aquellos ítems que puedan ser

de interés para los usuarios, utilizando la información disponible. Puesto que cada acción posible se corresponde con un ítem diferente en la colección, se ha optado por tomar los diferentes brazos como los posibles ítems a recomendar. Cada acción lleva asociada una recompensa inmediata que, trasladada al problema de los sistemas de recomendación, vendrá dada por la valoración $r(u, i)$. Asimismo, cada brazo tiene un valor acumulado, que representa cuán bueno cree el sistema que es el brazo, por tanto, es razonable que este valor corresponda con la estimación $\hat{r}(u, i)$. Este trabajo se centra en explorar la tarea de recomendación no personalizada empleando algoritmos de bandidos, por lo que la función recompensa, $\hat{r}(u, i)$ será reemplazada por $\hat{r}(i)$, es decir, el mismo valor para todos los usuarios. Finalmente, el usuario en los sistemas de recomendación será el elemento que proporciona una retroalimentación a la hora de accionar un brazo. Además se introduce la restricción de que un ítem no puede ser recomendado a un usuario en más de una ocasión. El papel del usuario es por tanto, indicar qué brazos pueden ser accionados en el bandido.

En la tabla 3.1 se muestra de manera resumida la correspondencia entre los elementos que forman parte de un sistema de recomendación clásico y los que forman parte de un algoritmo de bandidos.

Sistema de recomendación	**Bandidos multi-brazo**
ítem	brazo o acción
valoración promedio	valor de un brazo
valoración	recompensa
usuario	contexto

Tabla 3.1: Correspondencia entre sistemas de recomendación y bandidos multi-brazo

Habitualmente en los sistemas de recomendación, para cada usuario se devuelve un ranking con k ítems ordenados según la relevancia que el algoritmo estima para el usuario [Ricci et al., 2011]. Si bien es posible devolver un ranking empleando estrategias de bandidos [Zhao et al., 2013], en general, en la literatura, se aconseja recomendar un ítem en cada iteración, lo que se ajusta de mejor manera a la filosofía de los bandidos multi-brazo [Wang et al., 2018].

3.1.2. Recomendación como proceso iterativo

El proceso de recomendación con bandidos multi-brazo es interactivo y continuo, es decir, un usuario pide una recomendación al sistema, el sistema devuelve un ítem y a cambio obtiene una valoración, positiva o negativa, por parte del usuario, que utiliza para actualizar la información conocida hasta el momento. En el caso de no obtener ninguna valoración, se registra el ítem para no repetir la recomendación en un futuro, pero no se actualiza el valor del brazo.

En ocasiones se puede contar con un conjunto previo de valoraciones, que se emplea para inicializar el valor de los ítems ($\hat{r}(i)$) a la media de valoraciones entre todos los usuarios que lo hayan puntuado. De no contar con este conjunto, típicamente el valor de los brazos se inicializará a 0, aun-

que a lo largo de los sucesivos capítulos se explorarán diferentes formas de inicializar este valor. En este trabajo, se abordan dos formas de estimar r, ambas basadas en la recompensa promedio, que se detallan en las ecuaciones 3.2 y 3.3. Como se trata de un proceso iterativo, la recompensa asociada a cada ítem varía según se avanza en las épocas, por ello aparece el parámetro t para modelar esta variación temporal.

$$\begin{cases} \hat{r}_t(i) = \dfrac{\alpha_t(i)}{\alpha_t(i) + \beta_t(i)} = \dfrac{\sum_{u \in \hat{U}_t(i)} r(u,i)}{|\hat{U}_t(i)|} \\ \hat{r}_0(i) = c \end{cases} \tag{3.2}$$

donde $\hat{U}_t(i) = \{u_k : (u_k, i)$ escogida para $k \leq t\}$, es el conjunto de usuarios a los que se les ha recomendado el ítem i hasta el tiempo t, y $c \in [0,1]$, constante. En la fórmula 3.2 se estima el valor asociado a un ítem promediando el número de aciertos ($\alpha_t(i)$) entre el total de veces que el ítem ha sido escogido, o entre la suma de aciertos y fallos ($\alpha_t(i) + \beta_t(i)$).

$$\begin{cases} \hat{r}_t(i) = \dfrac{\hat{r}_{t-1}(i) + r(u_t, i)}{2} \\ \hat{r}_0(i) = c \end{cases} \tag{3.3}$$

donde u_t es el usuario a recomendar en tiempo t y $c \in [0,1]$ constante. En la ecuación 3.3 también se promedia el número de aciertos, pero en lugar de dividir entre el total de veces que un ítem ha sido seleccionado, se hace una media ponderada, dividiendo entre 2 en cada iteración, por lo que se otorga más peso a las últimas iteraciones. En ambas fórmulas, hay que inicializar el valor de cada ítem en $t = 0$.

A continuación, mediante el algoritmo 3.1 se muestra el proceso general por el que se lleva a cabo la recomendación a los usuarios del sistema.

```
input : U conjunto de usuarios, I conjunto de items
1 foreach u ∈ U do
2     i ← accionar_brazo(u);
3     r ← obtener_recompensa(u, i, R);
4     actualizar_brazo(i, r);
5 end
```

Algoritmo 3.1: Algoritmo general de recomendación con bandidos

El método para seleccionar un brazo vendrá determinado por el algoritmo de bandidos escogido. El brazo se actualiza según una de las fórmulas 3.2 o 3.3.

3.2. Algoritmos

Existen multitud de algoritmos de bandidos multi-brazo que pueden adaptarse a la recomendación no personalizada propuesta en este trabajo. Estos algoritmos se diferencian fundamentalmente en cómo se balancea la parte de exploración y la parte de explotación correspondiente. En este capítulo se describen los algoritmos de ε-Greedy [Sutton and Barto, 2018], *Upper Confidence Bound (UCB)* [Auer et al., 2002] y Thompson Sampling [Chapelle and Li, 2011], que, dentro de los muchos algoritmos que existen y las múltiples variantes que hay, son los que se han considerado para este trabajo.

Si bien se ha empleado un enfoque no personalizado, los algoritmos de bandidos multi-brazo permiten también un enfoque personalizado [Zhao et al., 2013, Wang et al., 2018, Kawale et al., 2015, Gentile et al., 2014, Li et al., 2010]. La diferencia fundamental entre estas dos estrategias, a nivel de bandidos, es que en la no personalizada, solo existe un único bandido en el que cada brazo tiene un mismo valor independientemente del usuario. Además en este trabajo los brazos aproximan la recomendación por el valor promedio. Para lograr un enfoque personalizado, cada brazo o ítem debería guardar un valor distinto para cada usuario, además de almacenar qué ítems ya han sido recomendados para cada usuario con el fin de no repetir las recomendaciones.

3.2.1. ε-Greedy

El algoritmo de ε-Greedy [Sutton and Barto, 2018] balancea la exploración y explotación de una manera muy sencilla: se fija un ε en el intervalo $[0, 1]$ y con probabilidad ε se selecciona un ítem o acción uniformemente al azar, lo que corresponde con la fase de exploración, y con probabilidad $1 - \varepsilon$ se selecciona el ítem que maximiza $\hat{r}(i)$, que sería la fase de explotación. Este comportamiento se resume en la fórmula 3.4.

$$i_t = \begin{cases} i \in I \text{ uniformemente al azar}, & 1 - \varepsilon \\ \underset{i \in I}{\operatorname{argmax}}[\ \hat{r}_t(i)\ ], & \varepsilon \end{cases} \tag{3.4}$$

El valor $\hat{r}_0(i)$ (ver fórmulas 3.2 o 3.3) se inicializará de la misma forma para todos los ítems. Típicamente se usará el valor 0, no obstante en algunos experimentos se dará un número inicial de aciertos (α_0) y fallos (β_0) y el valor de $\hat{r}_0(i)$ vendrá dado por la fórmula 3.5.

$$\hat{r}_0(i) = \frac{\alpha_0}{\alpha_0 + \beta_0} \tag{3.5}$$

Esta inicialización será referida en posteriores capítulos como “inicialización optimista/pesimista”. El propósito es suponer que en lugar de tener 0 aciertos y 0 fallos al inicio de la iteración, se tienen inicialmente α_0 aciertos y β_0 fallos. Si $\alpha_0 > \beta_0$ se supone que inicialmente el número de aciertos es mayor que el número de fallos, es decir, el valor inicial de todos los ítems es mayor que $0{,}5$, lo cual

se considera una inicialización optimista. Por otro lado, $\alpha_0 < \beta_0$ correspondería a una inicialización pesimista.

3.2.2. Upper Confidence Bound

La idea tras el algoritmo de Upper Confidence Bound es tomar la recomendación basada en popularidad, es decir, una estrategia codiciosa, y añadirle un término exploratorio. De esta forma, el algoritmo de UCB [Auer et al., 2002], selecciona en cada época el ítem que maximiza la fórmula 3.6

$$i_t = \underset{i \in I}{\operatorname{argmax}} \left[\hat{r}_t(i) + \sqrt{\frac{\gamma \, ln(t)}{\alpha_t(i) + \beta_t(i)}} \right] \tag{3.6}$$

El primer sumando en la ecuación 3.6 se refiere, como ya se ha visto, a la estimación del *reward* asociado a un ítem. De no existir el segundo sumando, se trataría de una estrategia codiciosa y trivial.

El segundo sumando se conoce como *uncertainty* o incertidumbre y modela la parte exploratoria del algoritmo. El parámetro γ es una constante que permite otorgar más peso a la incertidumbre. En el numerador, aparece la época en la que se encuentra el algoritmo, que se suaviza con el logaritmo. En el denominador se toma la suma de aciertos y fallos, equivalentemente, el número de veces que el ítem i ha sido seleccionado. De esta forma, cuanto más veces haya sido seleccionado un ítem, mayor será el denominador, y por tanto menor el valor de la incertidumbre, lo que intuitivamente implica que el algoritmo tiene mayor certeza del valor de $\hat{r}_t(i)$. Por el contrario, cuanto menos haya sido accionado un brazo, la incertidumbre será mayor y, puesto que se selecciona el ítem que maximiza esta cantidad, un denominador menor, favorecerá la exploración a los ítems poco seleccionados.

Para $t = 0$, el valor de la incertidumbre es indeterminado, por ello el algoritmo de UCB requiere de una primera vuelta de inicialización. En esta inicialización, se recomiendan todos los ítems al menos una vez a diferentes usuarios (se ha optado por recorrer los usuarios secuencialmente, pero se podrían explorar otras estrategias). Tras esta vuelta, t toma el valor de $|I|$, puesto que se ha forzado a seleccionar cada ítem una única vez; $\hat{r}_t(i)$ tomará valor 0 o 1 dependiendo de si la recomendación ha sido efectiva, al igual que $\alpha_t(i)$ y $\beta_t(i)$.

En la figura 3.1 se muestra la evolución del algoritmo de UCB para dos ítems al inicio, a mitad y al final de la ejecución. Los puntos rojos representan el valor promedio de cada ítem, mientras que los puntos azules son el resultado de sumar la incertidumbre al valor anterior. Inicialmente ambos ítems parten con un valor estimado 0. La incertidumbre inicial es la misma para todos los ítems y es el resultado de la vuelta exploratoria del algoritmo. Por tanto inicialmente la recomendación se hace de manera aleatoria. En las sucesivas iteraciones se observa como el valor de la incertidumbre se va reduciendo y por tanto los puntos azul y rojo se van acercando, conforme el ítem en cuestión se va recomendando. Se comprueba que el segundo ítem, el situado a la derecha en la figura, es más popular que el primero, no solo por tener un valor promedio más alto, sino porque los puntos azules y

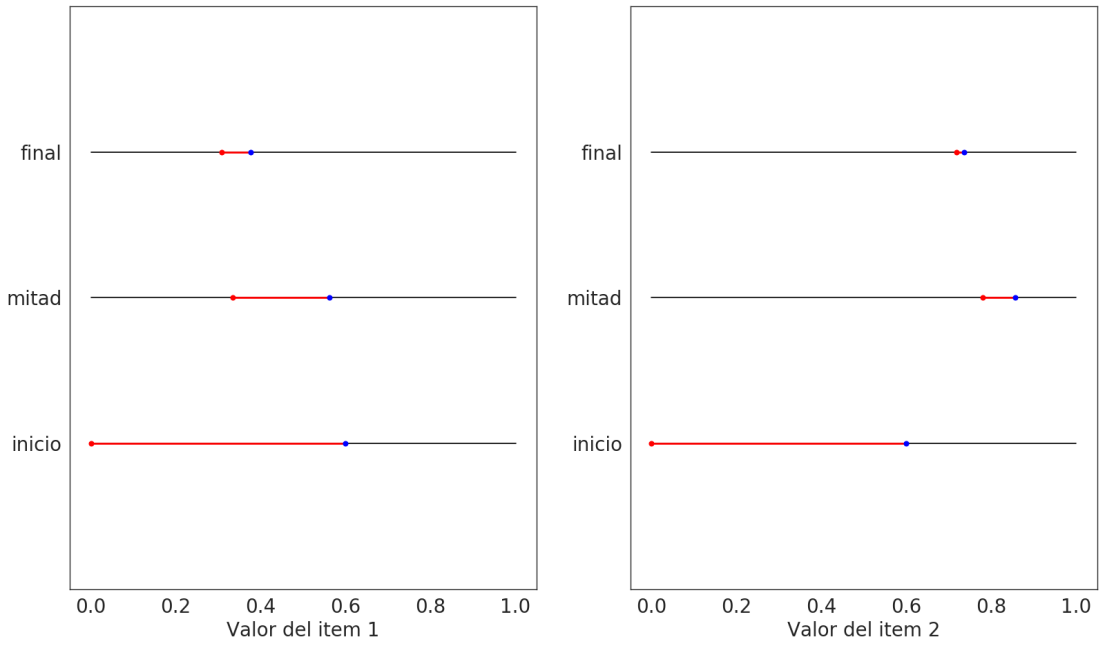

Figura 3.1: Comparación de la evolución de los valores de la estimación del *rating* y de la incertidumbre de dos ítems para el algoritmo de UCB

rojo se acercan más rápidamente, lo que quiere decir que el ítem se recomienda más veces.

3.2.3. Thompson Sampling

Una estrategia de bandidos debe descubrir –lo antes posible– el brazo que genera la recompensa promedio más alta. Una estrategia de explotación pura elegiría siempre el brazo con la media observada más alta hasta el momento. Las estrategias de bandidos añaden un ingrediente de exploración que se basa, de una manera distinta en cada método, en reconocer la incertidumbre que encierran las observaciones disponibles con respecto a una muestra parcial de la distribución de la recompensa en cada brazo. En ε-Greedy la incertidumbre se refleja de forma muy rudimentaria, introduciendo una tasa de elección aleatoria. En Thompson Sampling [Chapelle and Li, 2011], la incertidumbre se tiene en cuenta *muestreando* una media para cada brazo, en base a las recompensas observadas hasta el momento, en lugar de calcular una media directa con esas observaciones. Esa muestra se puede generar estimando una distribución de la media basada en los valores observados.

La función r (ver fórmula 3.1) se ha definido de manera que solo puede tomar valores discretos 0 o 1, por lo tanto, resulta razonable tomar la distribución de la media como una Bernoulli (una “moneda”) de parámetro p_i (la probabilidad de que al lanzar la moneda salga “1”). Cuando un brazo se ha escogido t veces, se contabilizan una serie de aciertos, $\alpha_t(i)$, que se corresponden con el número de veces que

se obtiene el valor 1, y fallos, $\beta_t(i)$, el número de veces que se observa el valor 0. El parámetro p_i puede, por tanto, describirse mediante una distribución Beta, la conjugada de una Bernoulli, con parámetros $\alpha_t(i)$ y $\beta_t(i)$. Cuantos más valores se han observado, es decir, cuanto mayor es $\alpha_t(i)+\beta_t(i)$ más se estrecha la distribución Beta en torno a su media, $\frac{\alpha_t(i)}{\alpha_t(i)+\beta_t(i)}$, y por tanto más certeza se tiene del valor asignado a cada brazo. Este comportamiendo se ilustra en la figura 3.2. Así pues, el algoritmo selecciona en cada época el ítem según la fórmula 3.7

$$i_t = \underset{i \in I}{\operatorname{argmax}}[x \leftarrow B(\alpha_t(i) + \alpha_0, \beta_t(i) + \alpha_0)] \tag{3.7}$$

donde $B(a, b)$ denota la distribución Beta de parámetros a, b y "$\leftarrow$" denota un número aleatorio. Generalmente los parámetros de la distribución Beta se inicializarán a 1, es decir, las distribuciones Beta comienzan siendo uniformes. No obstante también se pueden tomar mayores para la inicialización optimista ($\alpha_0 > \beta_0 \geq 1$) o pesimista ($\beta_0 > \alpha_0 \geq 1$).

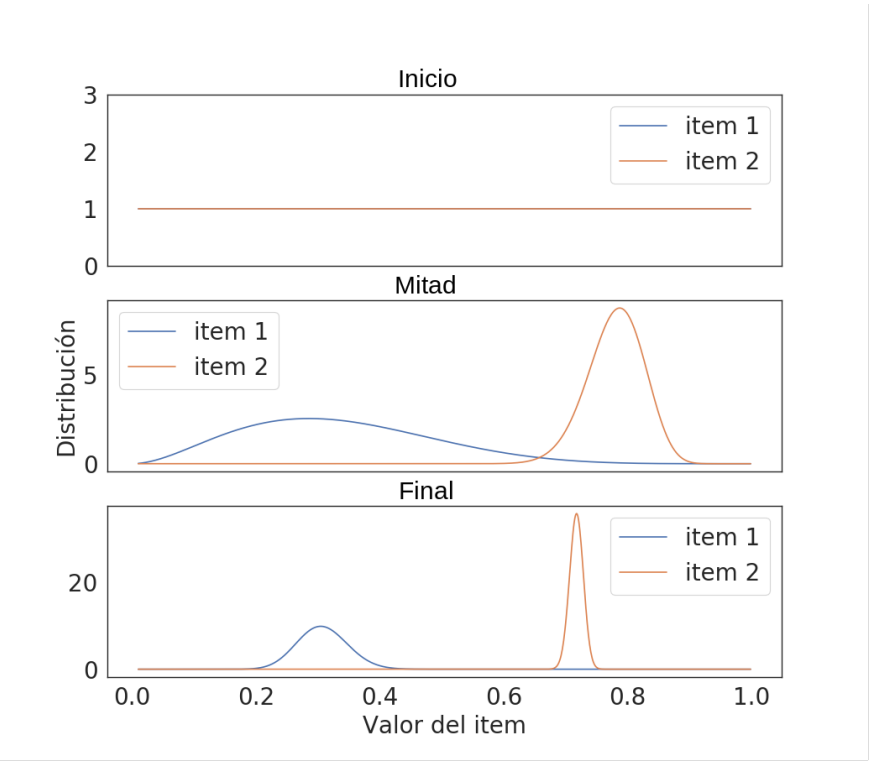

Figura 3.2: Comparación de la evolución de las distribuciones de dos items para el algoritmo de Thompson Sampling

En la figura 3.2 se puede observar la evolución de las distribuciones que siguen dos ítems durante la evolución del algoritmo de Thompson Sampling. Inicialmente ambos ítems parten con una distribución uniforme, en este caso se ha tomado para ambos $\alpha_0 = \beta_0 = 1$. En las siguientes subfiguras, se representan las distribuciones a la mitad de iteraciones y al finalizar la ejecución del algoritmo. Se observa como al avanzar en las iteraciones, la distribución se va centrando en torno al valor del ítem. Asimismo se comprueba que el segundo ítem es más popular que el primero, pues además de tener

un valor promedio mayor, la distribución se acerca más rápido al valor estimado del ítem debido a que se tiende a recomendar más frecuentemente.

4

EXPERIMENTOS

En este capítulo se describe un conjunto de experimentos realizados con los algoritmos de recomendación de bandidos descritos en el capítulo anterior (3.2). La finalidad es observar su comportamiento, comprobar su eficacia, y comparar la de unos y otros algoritmos entre sí. Para ello se han utilizado varios conjuntos de datos de dominio público de diferente naturaleza. Se explora asimismo la idoneidad de diferentes configuraciones de sus parámetros.

En la sección 4.1 se describen en detalle los experimentos que se realizarán en las sucesivas secciones y sus objetivos, así como la forma de evaluación de cada uno de ellos y las características principales de los conjuntos de datos escogidos para probar los algoritmos.

4.1. Metodología experimental

4.1.1. Descripción de los experimentos

Los experimentos de esta sección tienen dos propósitos principales. El primero de ellos es evaluar la efectividad de los algoritmos de 3.2 y compararlos con algoritmos clásicos de recomendación no personalizada. El segundo es comparar los tres algoritmos entre sí, explorando cuál es la mejor configuración de los parámetros de cada uno. Para ello se han realizado dos tipos de experimentos.

1. **Ajuste de parámetros.** Tanto el algoritmo de ε-Greedy como el de UCB dependen de un parámetro en su formulación. El objetivo de este experimento es determinar para qué valores de estos parámetros se obtiene un mejor resultado a la hora de realizar las recomendaciones. Para ello se hará una búsqueda en rejilla explorando diferentes valores de estos parámetros. Los resultados se mostrarán mediante dos tipos de gráficas distintas. La primera corresponderá a una comparación de las diferentes curvas de $recall$ acumulado obtenidas para cada valor del parámetro. La segunda es un corte a la mitad del número de iteraciones de la anterior.
2. **Inicialización optimista o pesimista.** En este experimento se evalúa como afecta una inicialización optimista o pesimista de los algoritmos. Como ya se adelantó en el capítulo 3,

el valor de $\hat{r}_0(i)$ puede inicializarse según la fórmula 3.5. Una inicialización optimista consiste en tomar en la ecuación 3.5 $\alpha_0 > \beta_0 > 0$, mientras que en la inicialización pesimista se toma $\beta_0 > \alpha_0 > 0$. Para realizar este experimento se tomará una muestra de 100 puntos para cada uno de los algoritmos tomando el parámetro del algoritmo, si lo tuviese, al mejor valor obtenido en el experimento anterior y variando α_0 y β_0 entre 0 y 10. Los resultados se mostrarán mediante un mapa de color al que se le aplicará una interpolación para suavizar los bordes.

4.1.2. Evaluación

Debido al carácter experimental de las pruebas realizadas, de los dos métodos de evaluación expuestos en la sección 2.1, se ha optado por una evaluación *offline*. No obstante, los algoritmos de bandidos plantean una gran ventaja respecto al arranque en frío (*cold start*) de los sistemas, ya que no es necesaria información previa sobre los gustos de los usuarios, puesto que las recomendaciones pueden comenzar con una fase exploratoria para obtener conocimiento y en las épocas posteriores explotar esa información. Para explorar mejor este hecho, el conjunto de *ratings* de entrenamiento se toma vacío, es decir, inicialmente el sistema no dispone de ninguna información acerca de los usuarios o los ítems y debe ir haciendo recomendaciones sucesivas hasta descubrir por completo el conjunto de test. La información obtenida al realizar las recomendaciones se va añadiendo al conjunto de entrenamiento con el objetivo de no recomendar más de una vez un ítem a un mismo usuario. Así, el algoritmo 3.1 se ve modificado como se muestra en el algoritmo 4.1.

input : U conjunto de usuarios, I conjunto de items, R_{train} conjunto de *ratings* de entrenamiento, R_{test} conjunto de *ratings* de test
output: $recall$ acumulado

```
1  recall ← 0;
2  while R_test ≠ ∅ do
3      foreach u ∈ U do
4          i ← accionar_brazo( u );
5          r ← obtener_recompensa( u, i, R );
6          actualizar_brazo( i, r );
7          eliminar_recompensa( i, u, R_test );
8          recall ← actualizar_recall( i, r );
9          actualizar( R_train );
10     end
11 end
```

Algoritmo 4.1: Algoritmo general de recomendación con bandidos con evaluación *offline*

La métrica escogida para evaluar las recomendaciones es el $recall$, que se calcula en cada época atendiendo a la fórmula 4.1 e indica la proporción del número de *ratings* positivos que se han descubierto en tiempo t

$$recall(t) = \frac{\sum_{k=1}^{t} r(u_k, i_k)}{|\{(u,i) : r(u,i) = 1, u \in U, i \in I\}|} \tag{4.1}$$

donde $u_k,\ i_k$ denota la pareja usuario-ítem seleccionada en la k-ésima época. En el denominador se cuentan el número de valoraciones positivas que hay inicialmente en el conjunto de test.

4.1.3. Descripción de los conjuntos de datos

Para realizar los experimentos descritos, se ha optado por utilizar tres conjuntos de datos de dominio público de diversa índole. A continuación se expone una breve descripción del tipo de ítem que se maneja en cada uno de los conjuntos y como se han tratado las puntuaciones de los usuarios a los ítems. En la tabla 4.1 se muestra la información del número de usuarios, ítems y *ratings* en cada conjunto de datos.

1. **CM100K.** Los datos de CM100K [Cañamares and Castells, 2018] contienen 103.584 puntuaciones anónimas sobre 1.054 canciones. Para simplificar el experimento y siguiendo la función definida en la ecuación 3.1, las puntuaciones se han binarizado, de manera que los items puntuados con 1 o 2 se consideran como 0 (al usuario no le gustó la canción) y los puntuados con 3 y 4 pasan a valer 1. Adicionalmente se dispone de un segundo dato que indica si el usuario ya conocía la canción antes de puntuarla.
2. **MovieLens1M.** El conjunto de datos de MovieLens1M [Harper and Konstan., 2015] contiene 1.000.209 valoraciones anónimas sobre aproximadamente 3 900 películas de 6 400 usuarios del sistema MovieLens [movielens, 1997]. Las valoraciones oscilan en una escala de 5 estrellas, que, al igual que con el conjunto de datos de CM100K, se han binarizado: de 1 a 3, se considera como 0 y de 4 a 5 como 1.
3. **Twitter.** Para el experimento sobre los datos de Twitter se cuenta con un conjunto de interacciones (*retweets* y menciones) de los últimos 200 *tweets* publicados por un conjunto de aproximadamente 10.000 usuarios [Sanz-Cruzado and Castells, 2018], sobre las que se ha tomado un subconjunto aleatorio de 5.000 usuarios. En Twitter se pueden recomendar tanto *tweets* como usuarios. En este caso solo se dispone de información de interacciones de un usuario a otro, por lo que se recomiendan usuarios basándose en las interacciones que estos tienen. Por tanto, se presenta la peculiaridad de que los ítems que se recomiendan son los propios usuarios. Además, se considera una restricción adicional: si a un usuario u_1 le sigue otro usuario u_2 pero no se da la relación recíproca, no se recomendará a u_1 seguir a u_2 puesto que se trata de una recomendación bastante trivial que incluso puede llegar a molestar a los usuarios. En este conjunto de datos no existe el *rating* de valor 0, puesto que no existe la opción de valorar negativamente a un usuario o *tweet*. Por este motivo todas las relaciones de interacción presentes entre los usuarios de la muestra se añaden al conjunto de test con valor 1.

Conjunto de datos	Número de usuarios	Número de ítems	Número de *ratings*
CM100K	1.000	1.054	103.584
MovieLens1M	6.400	3.900	1.000.209
Twitter	5.000	5.000	43.444

Tabla 4.1: Resumen de las principales características de los conjuntos de datos

4.2. Ajuste de parámetros: búsqueda en rejilla

El objetivo de este experimento es determinar qué valores de ε en para el algoritmo de ε-Greedy (ver fórmula 3.4) y qué valores de γ en el algoritmo de UCB (ver fórmula 3.6) proporcionan un mejor resultado a la hora de realizar las recomendaciones. Puesto que se ha utilizado como métrica el $recall$ (fórmula 4.1) un buen resultado se traduce en una curva de $recall$ de mayor pendiente. Para ello se ha utilizado una búsqueda en rejilla sobre los dos parámetros. Como el parámetro ε se trata de una probabilidad, solo puede tomar valores en el intervalo $[0, 1]$. No obstante, el parámetro γ puede tomar cualquier valor real mayor o igual que 0, por tanto, además de explorar valores próximos a 0, se exploran también valores grandes. Para todos estos experimentos el valor inicial para los brazos es 0 ($\alpha_0 = \beta_0 = 0$).

Los resultados se muestran mediante dos conjuntos de figuras. En la figura 4.1 se muestra una matriz de figuras en las que en cada columna se fija el algoritmo ε-Greedy y UCB, respectivamente, y en las filas se muestran los diferentes conjuntos de datos. En cada una de estas figuras se han dibujado las curvas de $recall$ para los distintos valores de los parámetros explorados. Para facilitar la comparación de los diferentes valores, en la figura 4.2, se muestra la misma configuración de matriz pero en cada subfigura se representa un corte de las gráficas de la figura 4.1 a mitad del número de iteraciones, de manera que se pueden comparar mejor los distintos valores de $recall$ obtenidos.

Un valor de ε cercano a 1 supone una mayor aleatoriedad en la recomendación, siendo $\varepsilon = 1$ la recomendación totalmente aleatoria, mientras que cuanto más se acerque su valor a 0, más codiciosa se vuelve la recomendación, lo que se traduce en recomendar el ítem del que se tiene constancia en el momento de ser el más popular. Recomendar el ítem más popular es una estrategia de recomendación no personalizada ampliamente utilizada, por tanto, las curvas con $\varepsilon = 0$ sirven para comparar la recomendación no personalizada empleando bandidos, con una estrategia no personalizada más tradicional. En todas las subfiguras se observa que la recomendación aleatoria produce una línea de $recall$ recta, ya que como se consideran recomendaciones binarias en todos los casos, la probabilidad de acertar es del 50 %, en cada época se habrán acertado aproximadamente la mitad de las recomendaciones hechas.

En los resultados obtenidos para el algoritmo de ε-Greedy, independientemente del conjunto de datos, se observa la tendencia de que los valores de ε más cercanos a 0 proporcionan un mejor

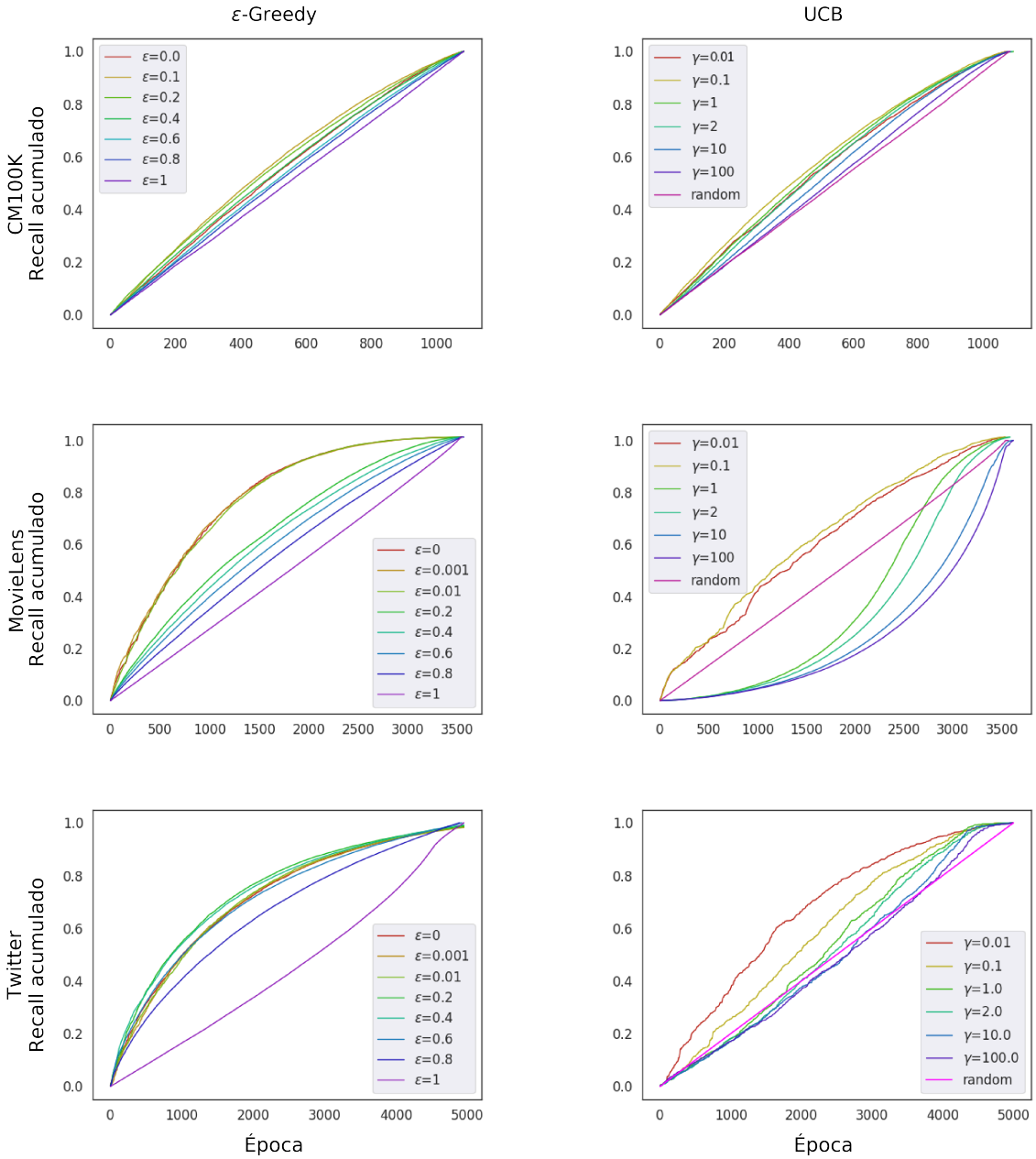

Figura 4.1: Resultados para la búsqueda en rejilla

resultado, lo cual era esperable puesto que se alejan de la recomendación aleatoria. Tanto en CM100K como en Twitter se observa cómo hay varios valores de ε que superan a la curva con $\varepsilon = 0$, lo que indica que un cierto grado de exploración es positivo para la recomendación y que por tanto para estos conjuntos de datos las estrategias de bandidos multi-brazo proporcionan un mejor resultado que la recomendación basada en popularidad.

En el caso de UCB, un valor de γ cercano a 0 supone un mayor grado de explotación. Cuanto más alto sea γ, más peso se le da a la incertidumbre y por tanto se favorece la recomendación de ítems que han sido poco recomendados (exploración). De nuevo se observa la tendencia de que cuanto menor se escoja parámetro γ, mejor resultado se obtiene. Al igual que con ε, tanto en CM100K como en Twitter, hay valores de γ para los que sus curvas de $recall$ superan a la recomendación basada en popularidad. El caso de MovieLens1M resulta curioso puesto que se observan curvas que están por debajo de la recomendación aleatoria. Esto se debe a que, si se observa la fórmula 3.6, cuanto más alto sea γ, más peso se le otorga al segundo sumando. En el denominador está presente el número de veces que un ítem ha sido recomendado mientras que el numerador es constante para todos los ítems. Por tanto, se tienden a recomendar los ítems con un denominador más pequeño, lo que se traduce en los ítems que han sido recomendados menos veces. Al observar la curva de $\varepsilon = 0$ en el conjunto de datos de MovieLens1M, se ve cómo la recomendación basada en popularidad supera a la mayoría de curvas, en gran medida, lo cual es indicativo de que en este conjunto de datos predominan unos ítems muy populares que le gustan a la gran mayoría de los usuarios. Sin embargo para las curvas con $\gamma > 1$, se van recomendando los ítems menos populares hasta que ya no se pueden recomendar más y van quedando cada vez ítems más populares. Esto explicaría por qué las estas curvas están por debajo de la recomendación aleatoria.

En la figura 4.2 se puede comprobar de manera más clara cuáles son los valores de ε y γ que proporcionan un mejor resultado. Puesto que la curva $recall$ de recomendación aleatoria se corresponde con la identidad, a la mitad de iteraciones se obtiene un valor de $recall = 0{,}5$. Para el algoritmo de ε-Greedy se obtienen los mejores resultados con $\varepsilon = 0{,}1$ para CM100K, $\varepsilon = 0$ (recomendación basada en popularidad) para MovieLens1M y $\varepsilon = 0{,}2$ para Twitter. De nuevo se concluye con estos valores que con ϵ cercano a 0 se obtienen mejores resultados. En el caso de UCB los mejores valores son $\gamma = 0{,}1$ para CM100K, $\gamma = 0{,}1$ para MovieLens1M y $\gamma = 0{,}001$ para Twitter. En el caso de MovieLens1M se concluye que la mejor estrategia de recomendación no personalizada es la basada en popularidad, mientras que para CM100K y Twitter si resulta positivo un cierto grado de exploración, especialmente para el segundo.

4.3. Inicialización optimista o pesimista

Esta sección tiene como objetivo determinar, para cada uno de los algoritmos de bendidos multi-brazo expuestos en la sección 3.2, si una inicialización optimista o pesimista de los valores iniciales de los

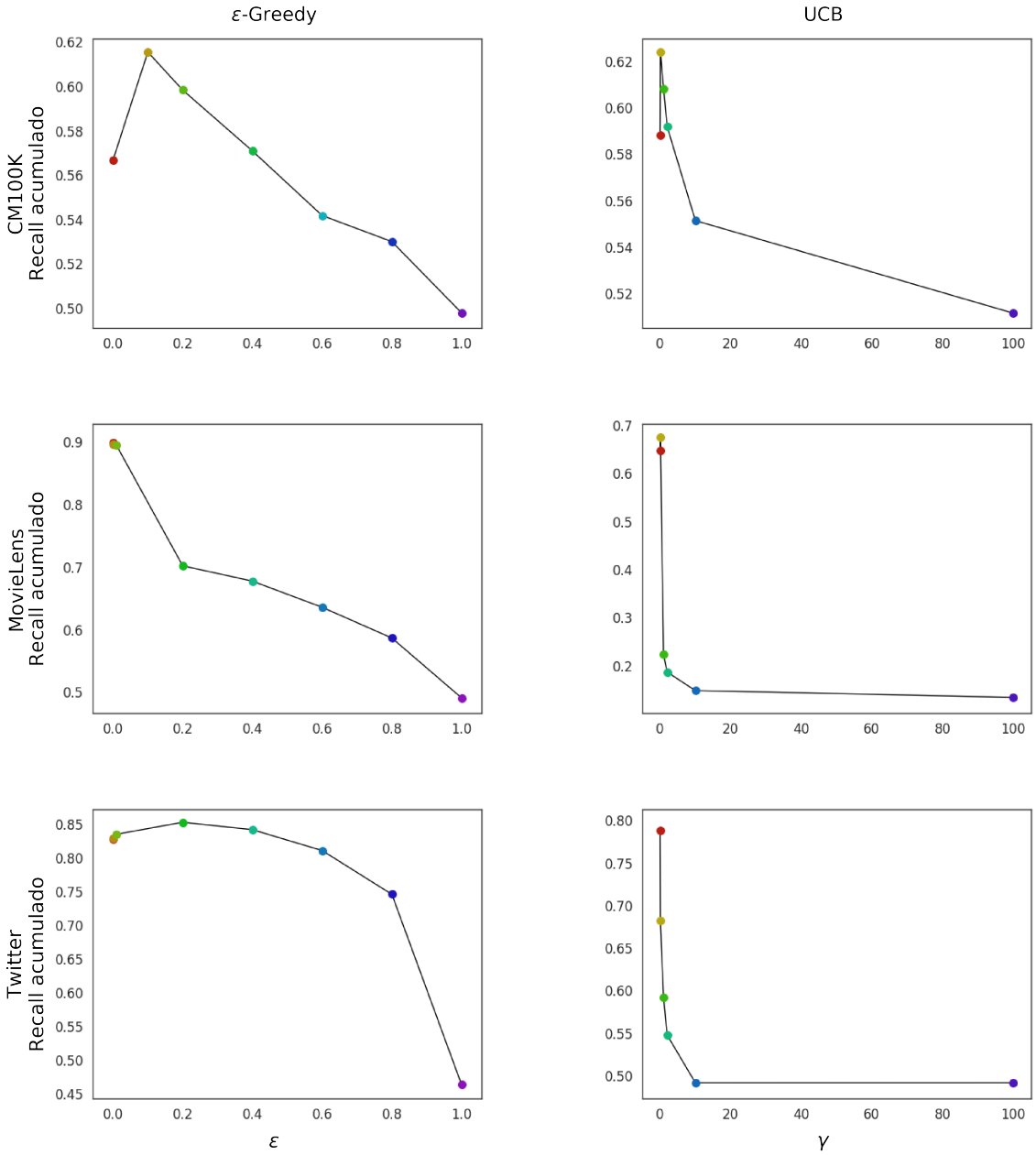

Figura 4.2: Corte a mitad del número de épocas de la figura4.1

brazos proporciona un mejor resultado. Para ello, para cada algoritmo y conjunto de datos se obtienen 100 curvas de $recall$ variando α_0 y β_0 de 0 a 10 con saltos de 1. Los resultados se presentan mediante un mapa de color para cada combinación de algoritmo y conjunto de datos, lo que supone un total de 9 figuras. En cada una de ellas para cada punto (α_0, β_0) se representa el valor que alcanza el $recall$ a la mitad del número de iteraciones. Los resultados pueden verse en la figura 4.3 donde en las columnas se fijan los conjuntos de datos y en la filas los algoritmos. En el caso de ε-Greedy y UCB, los parámetros ε y γ se fijan al mejor valor obtenido en el experimento anterior (sección 4.2).

Lo primero que llama la atención de la figura 4.3, es la gran variabilidad observada en todos los mapas de color de ε-Greedy, donde continuamente se alternan zonas azules y rojas. Si bien en el conjunto de datos de CM100K se observa una zona de $recall$ bajo entorno a $\alpha_0 = 7$ para los otros dos conjuntos de datos no se observa ningún patrón claro. Además, al fijarse en la barra de color, se observa que las variaciones que sufre el $recall$ son muy pequeñas, por lo que puede deberse a ruido al realizar los experimentos. Por tanto se concluye que las inicializaciones optimistas o pesimistas no afectan considerablemente al rendimiento del algoritmo de ε-Greedy.

El comportamiento de UCB y Thompson Sampling es muy similar en los tres conjuntos de datos, en todos ellos se observa de manera clara como las estrategias pesimistas ($\alpha_0 < \beta_0$) conducen a un mejor resultado. En el caso de CM100K, la frontera que separa las zonas azul y roja está en torno a la recta $\alpha_0 = \beta_0$. Para este conjunto de datos el mejor valor de UCB se obtiene en la combinación más pesimista de las exploradas, con $\alpha_0 = 1$ y $\beta_0 = 10$ y el peor resultado se obtiene en la más optimista, con $\alpha_0 = 10$ y $\beta_0 = 1$. Con Thompson Sampling el resultado es menos extremo, el punto de máximo $recall$ se encuentra en $\alpha_0 = 1$ y $\beta_0 = 8$ y el de mínimo en $\alpha_0 = 8, \beta_0 = 2$. Los valores de Thompson Sampling son en general superiores a los de UCB.

En el caso de MovieLens los valores obtenidos con UCB son bastante inferiores a los del resto de los algoritmos. Para Thompson Sampling se observa como las zonas azul y roja son aproximadamente del mismo tamaño, alcanzando el máximo en $\alpha_0 = 1, \beta_0 = 10$ y el mínimo en $\alpha_0 = 10, \beta_0 = 1$. En el caso de UCB la zona roja es casi imperceptible y el máximo se alcanza en valores más equilibrados $\alpha_0 = 2, \beta_0 = 2$. Se presentan asimismo, varios máximos locales en torno a la recta $\alpha_0 = \beta_0$.

Para los datos de Twitter, de nuevo se observa un comportamiento muy similar entre UCB y Thompson Sampling. Las fronteras entre las zonas azul y roja son en ambos casos una línea prácticamente vertical, siendo la de UCB en torno al valor de $\alpha_0 = 5$ y la de Thompson Sampling para $\alpha_0 = 4$. Los valores de Thompson Sampling son, al igual que con los demás conjuntos de datos superiores a los de los otros dos algoritmos. Tanto en UCB como en Thompson Sampling el mínimo se encuentra para $\beta_0 = 5$, aunque el de UCB es en una configuración más optimista que en Thompson Sampling, con $\alpha_0 = 10$ y $\alpha_0 = 7$ respectivamente. Para los dos algoritmos el máximo se alcanza para $\alpha_0 = 1$, siendo el caso de UCB, $\beta_0 = 1$, más optimista que el de Thompson Sampling, $\beta_0 = 7$. En general se puede concluir que existe una tendencia a que las configuraciones pesimistas conducen a un mejor resultado que las optimistas.

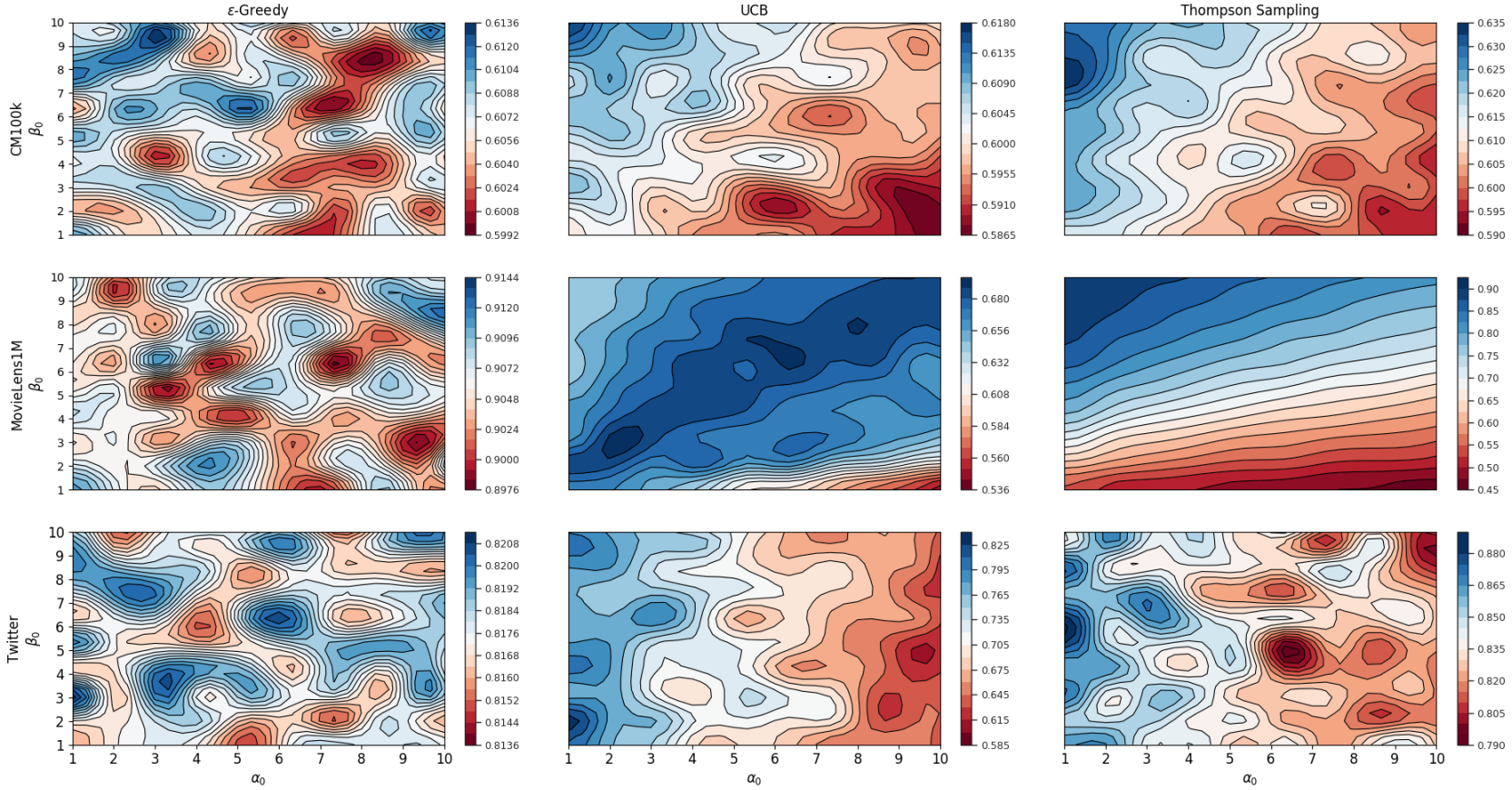

Figura 4.3: Resultados para la inicialización optimista o pesimista

4.4. Comparativa de resultados

Una vez se han explorado los parámetros de cada uno de los algoritmos en las secciones 4.2 y 4.3 y se ha determinado para qué valores de los parámetros se obtienen mejores resultados, en esta sección se comparan, para cada conjunto de datos, las mejores configuraciones de los tres algoritmos, con el objetivo de determinar cuál de ellos tiene un mejor rendimiento. Para facilitar la visualización de los resultados, a las curvas de $recall$ se le resta la curva de $recall$ aleatoria para aumentar las diferencias entre sus valores. Los resultados se muestran en la figura 4.4.

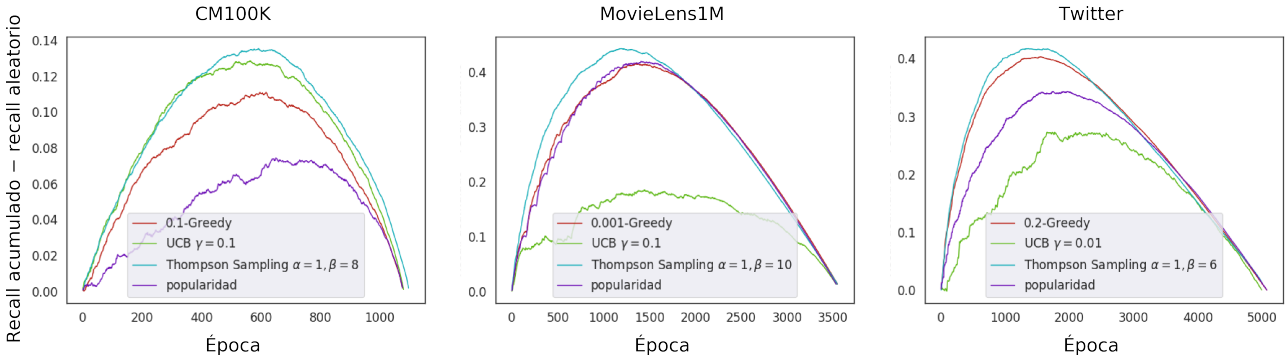

Figura 4.4: Comparativa de los resultados obtenidos para las mejores configuraciones de los algoritmos en los tres conjuntos de datos

A la vista de los resultados de la figura 4.4 se concluye que el algoritmo de Thompson Sampling es en general bastante robusto, proporcionando en todos los conjuntos de datos un resultado consistente y bueno, superando en todos los casos, no solo a los otros algoritmos de bandidos, sino también al algoritmo de popularidad. Le sigue el algoritmo de ε-Greedy, que pese a ser el más sencillo, obtiene unos resultados satisfactorios en todos los conjuntos de datos y supera en todos los casos excepto en MovieLens1M, a la curva generada por la recomendación basada en popularidad (ε = 0). Con respecto a UCB, los resultados obtenidos son, en general malos, a excepción de en CM100K, lo que indica una fuerte dependencia del conjunto de datos que se trata.

4.5. Acierto novedoso

Como ya se adelantó en la sección 4.1.3, para el conjunto de datos de CM100K se cuenta con un dato adicional que indica si el usuario ya conocía con anterioridad la canción que valoró. Este experimento tiene como objetivo utilizar esta información para medir el grado de novedad que consiguen las recomendaciones, es decir, a la hora de calcular el $recall$ se contabilizan las recomendaciones acertadas *y desconocidas* para el usuario y contrastar qué algoritmos son más efectivos cuando se trata de recomendar nuevo contenido a los usuarios. Para ello, se obtienen las curvas de $recall$ de cada uno de los tres algoritmos con la mejor configuración obtenida en el conjunto de datos de CM100K, es

decir $\varepsilon = 0{,}1$ para ε-Greedy, $\gamma = 0{,}1$ para UCB y $\alpha_0 = 1, \beta_0 = 8$ para Thompson Samplig y de nuevo se les resta la curva de $recall$ aleatoria para aumentar las diferencias entre sus valores y facilitar la visualización. Los resultados se muestran en la figura 4.5.

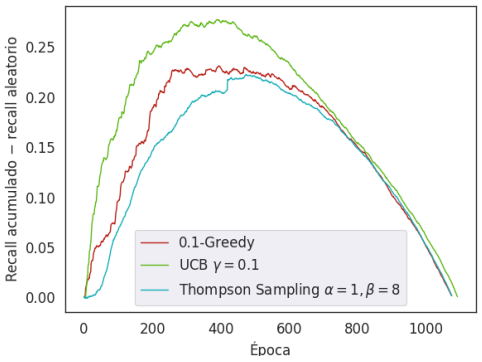

Figura 4.5: Resultados del $recall$ no descubierto

Al comparar los resultados de la figura 4.5 con los de la figura 4.4 correspondientes a CM100K, se observa que, ahora UCB pasa a ser el que mejor resultado obtiene, manteniendo el rendimiento obtenido con el $recall$. El resultado para ε-Greedy también se mantiene. No obstante, se observa como el acierto de Thompson Sampling disminuye mucho con respecto a los otros algoritmos. Esto es indicativo de que pese a que Thompson Sampling tiene un buen rendimiento en cuanto al acierto, no es muy efectivo descubriendo nuevo contenido a los usuarios.

5

Conclusiones

Este capítulo se dedica a concluir el documento, exponiendo en primer lugar un resumen de las tareas desarrolladas en este trabajo, destacando las contribuciones más importantes y, en segundo lugar, una lista de posibles mejoras y ampliaciones al trabajo desarrollado.

5.1. Resumen y contribuciones

En este trabajo se ha estudiado la perspectiva del aprendizaje por refuerzo aplicada a los sistemas de recomendación profundizando en los algoritmos de bandidos multi-brazo. En concreto se han escogido tres algoritmos de bandidos multi-brazo, ε-Greedy, Upper Confidence Bound y Thompson Sampling y se han implementado en Python para realizar recomendaciones no personalizadas. Con el fin de implementar estos algoritmos, se ha estudiado la tarea de recomendación para obtener una visión global y comprender la diferencia fundamental entre los algoritmos clásicos de recomendación y la perspectiva del aprendizaje por refuerzo. Por otro lado se ha estudiado el aprendizaje por refuerzo a rasgos generales y sus diferentes formalizaciones, profundizando en los métodos de bandidos multi-brazo. Una vez que se han comprendido los dos elementos centrales de este trabajo se ha adaptado la tarea de recomendación a la perspectiva del aprendizaje por refuerzo, estableciendo una correspondencia entre los elementos propios de los sistemas de recomendación y los algoritmos de bandidos.

Para evaluar la eficacia de estos métodos se han realizado dos tipos de experimentos. El primero tiene como objetivo determinar qué configuración de los parámetros de los algoritmos de ε-Greedy y UCB proporcionan los mejores resultados. Para ello se han tomado conjuntos de datos de diferentes dominios, en concreto, CM100K, de música; MovieLens1M, de películas y Twitter, red social en la que se recomiendan contactos; y se ha realizado una búsqueda en rejilla de los parámetros comparando las curvas de $recall$ obtenidas. La conclusión de este experimento es que el parámetro idóneo depende del conjunto de datos escogido, pero existe una tendencia general a que los parámetros que favorecen la explotación proporcionan un mejor resultado, siempre que se deje un pequeño margen para la exploración, frente a una estrategia codiciosa pura que solo explota el mejor ítem conocido hasta el momento. El segundo experimento se centra en explorar la inicialización de los valores que

toman los brazos. Para realizarlo se han tomado los mismos conjuntos de datos y se ha evaluado el $recall$ alcanzado a la mitad de las iteraciones diferentes con inicializaciones con mayor o menor grado de optimismo (valores cercanos a 1) o pesimismo (valores cercanos a 0). Tras este experimento se concluye de nuevo que los resultados dependen del conjunto de datos, aunque se aprecia una tendencia clara a que las estrategias pesimistas conducen a mejores resultados, sobre todo en los casos de Thompson Sampling y UCB. Asimismo se han comparado las mejores configuraciones de los algoritmos en cada uno de los conjuntos de datos concluyendo que el más robusto es el de Thompson Sampling, seguido por ε-Greedy. El algoritmo de UCB parece depender mucho del conjunto de datos escogido, obteniendo en algunos casos malos resultados.

Por último se ha explorado la capacidad que tienen los algoritmos de mostrar a los usuarios nuevo contenido que les resulte satisfactorio en el conjunto de datos de CM100K. Se concluye que los algoritmos de UCB y ε-Greedy obtienen un nivel de satisfacción alto cuando se trata de recomendaciones novedosas para el usuario, sin embargo el de Thompson Sampling obtiene peores resultados. La conclusión general de este trabajo es que en los dominios estudiados, los algoritmos de bandidos multi-brazo aplicados a la recomendación no personalizada proporcionan mejores resultados que los algoritmos clásicos dedicados a esta tarea, siempre que sus parámetros se ajusten correctamente.

5.2. Trabajo futuro

En este trabajo se han implementado tanto una plataforma de recomendación iterativa como varios algoritmos de bandidos multi-brazo. El objetivo del trabajo es estudiar el comportamiento de estos algoritmos en la tarea de recomendación y por ello el código desarrollado es muy sencillo y no se ha profundizado en conseguir una gran eficiencia computacional. Se plantea, por tanto, mejorar el código desarrollado, mejorando su eficiencia y formalizando los diferentes módulos desarrollados.

Tanto el algoritmo de ε-Greedy como el de UCB cuentan en la literatura [Kuleshov and Precup, 2014] con variaciones cuyo objetivo es dedicar las primeras épocas de ejecución a la exploración e ir aumentando paulatinamente el grado de explotación. Estas nuevas estrategias no han sido exploradas en este trabajo y podrían implementarse en un futuro para compararlas con las ya desarrolladas. Asimismo solo se ha considerado la perspectiva no personalizada de la recomendación por lo que se plantea abordar la recomendación personalizada con bandidos para el futuro así como su comparación con algoritmos clásicos que resuelven esta tarea como KNN o factorización de matrices. En esta línea se puede comprobar la capacidad de exploración que tienen los bandidos multi-brazo. Para ello se escogerían algoritmos de recomendación supervisada previamente entrenados con diferente número de *ratings* en el entrenamiento y se compararían con algoritmos de bandidos sin entrenar para determinar en qué punto el bandido no es capaz de superar a los algoritmos supervisados.

Por último, en este trabajo solo se han realizado experimentos en los que los bandidos no tienen

información sobre los usuarios y los ítems a priori. Es natural, por tanto, considerar explorar diferentes estrategias de calentamiento para los algoritmos, ya sean aleatorias, por popularidad o incluso procedentes de otro algoritmo de recomendación y estudiar como estas influyen sobre el acierto.

Bibliografía

[Auer et al., 2002] Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multi-armed bandit problem. *Machine learning*, 47(2-3):235–256.

[Cañamares and Castells, 2018] Cañamares, R. and Castells, P. (2018). Should I Follow the Crowd? A Probabilistic Analysis of the Effectiveness of Popularity in Recommender Systems. In *41st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018)*, pages 415–424, Ann Arbor, Michigan, USA.

[Chapelle and Li, 2011] Chapelle, O. and Li, L. (2011). An empirical evaluation of thompson sampling. In *25th Conference on Neural Information Processing Systems (NIPS 2011)*, pages 2249–2257, Granada, Spain.

[Galbraith, 2016] Galbraith, B. (2016). Python library for Multi-Armed Bandits.

[Gentile et al., 2014] Gentile, C., Li, S., and Zappella, G. (2014). Online clustering of bandits. In *31st International Conference on Machine Learning (ICML 2014)*, pages 757–765, Beijing, China.

[Harper and Konstan., 2015] Harper, F. M. and Konstan., J. A. (2015). The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19*.

[Hill et al., 2017] Hill, D. N., Nassif, H., Liu, Y., Iyer, A., and Vishwanathan, S. (2017). An efficient bandit algorithm for realtime multivariate optimization. In *23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2017)*, pages 1813–1821, Halifax, NS, Canada,.

[Kawale et al., 2015] Kawale, J., Bui, H. H., Kveton, B., Tran-Thanh, L., and Chawla, S. (2015). Efficient Thompson Sampling for Online Matrix-Factorization Recommendation. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 1297–1305. Curran Associates, Inc.

[Kuleshov and Precup, 2014] Kuleshov, V. and Precup, D. (2014). Algorithms for multi-armed bandit problems. *arXiv preprint arXiv:1402.6028*.

[Li et al., 2010] Li, L., Chu, W., Langford, J., and Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *19th international conference on World wide web (WWW 2010)*, pages 661–670, Raleigh, North Carolina, USA.

[Littman et al., 1995] Littman, M. L., Dean, T. L., and Kaelbling, L. P. (1995). On the complexity of solving markov decision problems. In *11th conference on Uncertainty in Artificial Intelligence (UAI 1995)*, pages 394–402, Montreal, Canada.

[movielens, 1997] movielens (1997). Sitio web MovieLens.

[Ouyang, 2017] Ouyang, Y. (2017). R package for Multi-Armed Bandit Simulation Study.

[Ricci et al., 2011] Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B. (2011). *Recommender systems handbook*. Springer, New York; London.

[Sanz-Cruzado and Castells, 2018] Sanz-Cruzado, J. and Castells, P. (2018). Contact recommendations in social networks. In Berkovsky, S., Cantador, I., and Tikk, D., editors, *Collaborative Recommendations: Algorithms, Practical Challenges and Applications*, pages 519–570. World Scientific Publishing.

[Scott, 2015] Scott, S. L. (2015). Multi-armed bandit experiments in the online service economy. *Applied Stochastic Models in Business and Industry*, 31(1):37–45.

[Shani et al., 2005] Shani, G., Heckerman, D., and Brafman, R. I. (2005). An mdp-based recommender system. pages 453–460, Edmonton, Alberta, Canada.

[Sutton and Barto, 2018] Sutton, R. and Barto, A. (2018). Reinforcement learning: An introduction (2nd ed). *MIT Press, Cambridge, MA, USA*.

[Wang et al., 2018] Wang, Q., Zeng, C., Zhou, W., Li, T., Iyengar, S. S., Shwartz, L., and Grabarnik, G. (2018). Online interactive collaborative filtering using multi-armed bandit with dependent arms. *IEEE Transactions on Knowledge and Data Engineering*.

[Zhao et al., 2013] Zhao, X., Zhang, W., and Wang, J. (2013). Interactive collaborative filtering. In *Proceedings of the 22nd ACM international conference on Conference on information; knowledge management*, CIKM '13, pages 1411–1420, New York, NY, USA. ACM.

DEFINICIONES

Aprendizaje supervisado Rama de la Ingeniería Informática dedicada al desarrollo de algoritmos y modelos estadísticos cuyo objetivo es aprender una función que transforma una entrada en una salida predefinida.

arranque en frío Del inglés *cold start*, es la situación en la que se encuentra un sistema de recomendación cuando inicialmente no dispone de información acerca de los usuarios o ítems que lo componen para realizar las recomendaciones.

búsqueda en rejilla Del inglés *grid search*, es un método de optimización de parámetros que consiste en fijar un conjunto de posibles valores para los parámetros y probar todas las combinaciones posibles sobre ellos.

cluster Conjunto o agrupación.

exploración En el contexto del aprendizaje por refuerzo, supone sacrificar el acierto inmediato realizando una acción no óptima para ganar conocimiento sobre la misma.

explotación En el contexto del aprendizaje por refuerzo, supone realizar la mejor acción conocida hasta el momento maximizando la recompensa inmediata.

Information Retrieval Rama de la Ingeniería Informática que abarca todas las técnicas y herramientas relacionadas con la búsqueda de información.

ítem Elemento a recomendar por un sistema de recomendación. Por ejemplo, canciones en plataformas como Spotify, vídeos en YouTube o en redes sociales como Twitter, tanto usuarios como publicaciones (*tweets*).

ranking Lista de ítems ordenados de mayor a menor según la relevancia para el usuario que estima el sistema de recomendación.

rating Puntuación otorgada por un usuario a un ítem en un sistema de recomendación.

test A/B Método de evaluación que permite evaluar dos sistemas, o dos variantes de un sistema, a partir de las interacciones de los usuarios con los mismos.

ACRÓNIMOS

KNN K-Nearest Neighbours.

MAB Multi-armed bandits.

MDP Markov Decision Process.

UCB Upper Confidence Bound.